제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

효율적인 데이터 관리를 위한 레벨-단위 데이터 분할 프로토콜

송성근*, 윤희용*, 이보경**, 최중섭**, 박창원**, 이형수**
*성균관대학교 정보통신 공학부
**한국 정보 보호 진흥원 정보 보호 기술팀
**전자 부품 연구원 정보시스템 연구센터
e-mail : *{songsk, youn}@ece.skku.ac.kr, **{bklee, jschoi}@kisa.or.kr,
**{parkcw, hslee}@keti.re.kr

Level-wise Information Dispersal Protocol for Efficient Data Management

Sung Keun Song*, Hee Yong Youn*
Bo Kyoung Lee**, Joong Sup Choi**
Park Chang Won**, Lee Hyung Soo**
*School of Information and Communication Engineering, Sungkyunkwan University
**Information Security Technology Division, Korea Information Security Agency
** IT System Research Center, Korea Electronics Technology Institute

요 약

서바이벌 스토리지 시스템(Survivable Storage System)은 데이터의 가용성 및 보안성을 높이기 위해 여러 가지 분할 복제 기법들을 사용한다. 이러한 기법들을 정보의 중요도를 고려하지 않고 모든 데이터에 일괄적으로 적용하면, 시스템의 성능면에서 비효율적이다. 본 논문은 이를 해결하기 위해 정보의 중요도별로 다른 정보 분할 기법(IDS : Information Dispersal Scheme)를 적용하는 레벨 단위 데이터 분할 프로토콜을 제안하고 그 성능을 평가한다. 그 결과 제안된 방식은 정보의 중요도가 높을수록 데이터의 실질적인 가용성 및 보안성을 증가시킨다는 것을 볼 수 있다.

1. 서론

오늘날 컴퓨터 네트워크 기술 및 인터넷의 급속한 발달과 더불어 사회의 정보화가 가속화되면서 국가기관이나 기업체에서 관리 및 보관해야 하는 정보의 양이 급격히 증가하게 되었으며, 이로 인해 정보들의 효율적인 관리 및 안전성 문제가 대두되었다. 이를 해결하기 위해 국내외적으로 Survivable Storage System [2]에 대한 연구가 활발이 이루어지고 있는 추세다.

분할 복제 기법이란 Survivable Storage System에서 시스템의 성능, 데이터의 가용성 및 안전성 측면에서 데이터를 분할, 복제하는 기법으로 여러 기법들이 연구, 개발되고 있다[1]. 대표적으로 복제(Replication), 스플리팅(Splitting), 데시메이션(Decimation), 정보 분할(Information Dispersal), 비밀 분할(Secret Sharing) 등이 있는데 이러한 기법의 적용에 있어서 데이터의 중요도 차이를 고려하지 않은 경향이 있다. 국가기관이나 기업체에서 관리 및 보관해야 하는 데이터마다 중요도에 차이가 있다. 이런 사실을 경시하고 모든 데이터마다 같은 기법을 적용하면 시스템의 성능이 저하될 뿐더러 비효율적이다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해 Survivable Storage System의 3가지 관점인[4, 6] 가용성, 안전성, 시스템 성능 측면에서 정보의 중요도를 고려한 레벨 단위 정보 분할 프로토콜을 구현하고자 한다.

본 논문에서 제안하는 레벨 단위 정보 분할 프로토콜은 정보 분할 기법(IDS : Information Dispersal Scheme)을 이용한 방식으로, 이 프로토콜은 레벨이 다른 데이터 마다 일정한 형식의 읽기(Read) / 쓰기(Write)을 적용하지만, 적용되는 IDS가 다르다. 그리고 데이터의 레벨이 증가 할수록 시스템에서의 데이터의 실질적인 가용성 및 안전성이 증가되도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 IDS와 IDS의 일반적인 특징에 대하여 간략히 알아보고, 3장에서는 레벨 단위 정보 분할 프로토콜의 구현에 대해 설명한다. 4 장에서는 구현된 프로토콜

을 여러 측면에서 데이터의 Read 및 Write의 가용성(availability)을 측정하고 데이터의 레벨마다 가용성의 차이가 어떻게 나는지 확인한다. 마지막으로 5 장에서는 결론을 맺는다.

2. IDS

(m, n)-IDS이란 데이터 M을 n개의 데이터 조각으로 분할해서 n개의 서버에 Write하고, Read할 때는 최소한 m개의 데이터 조각이 필요로 하는 기법이다. 이 기법은 Read시 n - m개의 서버 실패를 묵인할 수 있는 기법이다. 이 기법의 가용성은 다음의 식에 의해 구할 수 있다[6, 7].

$$P(m, n) = \sum_{i=m}^{n} \binom{n}{i} P_s^i (1-P_s)^{n-i} \qquad (1)$$

이런 (m, n)-IDS의 가용성에 영향을 미치는 3가지 요소가 있다[3].

- n : 정보의 분할 정도(Information dispersal degree)
- n/m : 정보의 확장율(Information expansion ratio)
- P_s : 서버의 성공확율(Server survivability)

IDS기법에서 각 클래스(Class)는 확장율은 같고 정보의 분할 정도가 다른 IDS들로 이루어진다. 일반적인 네트워크 시스템을 IDS기법으로 나타내면 (1, 1)-IDS이다. 즉, 서버의 성공 확률이 시스템의 가용성이 된다. 비슷한 방법으로 (m, m)-IDS의 가용성을 구하면 다음과 같다.

$$P(i, i) = P_s^i,\ i = 2, \cdots, m$$

확장율이 1인 Class의 임의의 두 IDS를 다음과 같이 비교해보면,

$$P(m, m) - P(n, n) = P_s^m - P_s^n$$
$$= P_s^m (1 - P_s^{n-m}) > 0 \ \ if\ m < n$$

이다. 이 식에서 알 수 있듯이 P_s가 고정된 상태에서 m값이 증가 할수록 가용성이 떨어지는 것을 알 수 있다. 또한, 확장율이 1인 상태에서 정보의 분할 정도가 증가해도 가용성은 개선되지 않는다는 것을 알 수 있다. 일반적으로 두개의 IDS가 있을 때 확장율이 큰 IDS가 가용성이 좋다. 그러나 모든 경우가 그런 것만은 아니다.

IDS에 대한 몇 가지 특성에 대해 알아보자.

- 정리1

$P(i, j) \le P(k, l)$ if $l \ge j$ and $k \le i$

- 정리2

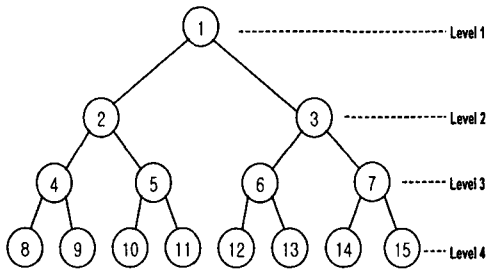
$P(i, j) > P(k, l)$ if $l \ge j$, $k > i$ and $l - k \le j - i$

정리1에서는 큰 데이터 조각들이 많은 서버에 저장될수록 데이터의 가용성이 좋다는 것을 알 수 있다. 정리2에서는 $l \ge j$, $k > i$ 이면서 서버의 실패를 묵인할 수 있는 서버의 대수가 많은 IDS가 가용성이 좋다는 것을 알 수 있다[3].

3. 제안하는 레벨 단위 데이터 분할 프로토콜

한 시스템에 저장|관리되는 데이터마다 정보의 중요도가 있다. 레벨 단위 데이터 분할 프로토콜은 이러한 데이터의 중요도에 입각하여 중요도가 서로 다른 데이터들마다 다른 IDS를 적용하는 방식이다. 다시 말해, 데이터의 중요도가 높을수록 가용성이 높은 IDS를 적용한다.

레벨 단위 데이터 분할 프로토콜은 네트워크 구조로 자식 노드 수가 일정한 논리적인 트리 구조를 이용한다. (그림 1)은 자식 노드수가 2인 트리 구조의 예를 들고 있다. 그리고 보는 바와 같이 트리의 각 단계별로 레벨이 정해져 있다.

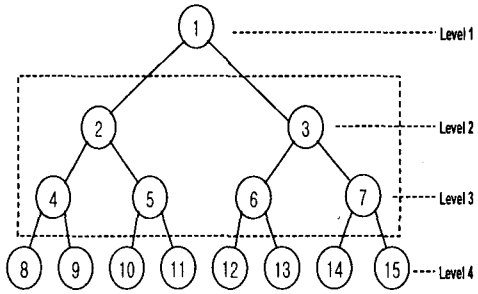


(그림 1) 자식 노드수가 2인 트리 구조.

각 데이터 마다 중요도 즉, 레벨이 정해진 상태에서 임의의 데이터를 Write 할 경우 Write 알고리즘은 다음과 같다.

- 레벨이 1인 데이터는 루트 노드에만 저장한다. 레벨이 2 이상이면 다음 단계를 적용한다.
- 먼저 Write할 데이터의 레벨과 일치하는 트리의 레벨의 노드 수만큼 데이터를 분할 저장한다.
- 그 다음 분할 저장된 데이터마다 한 레벨 위의 자식 노드 수만큼 복제한 후 그 자식 노드에 저장한다.

(그림 2)는 레벨이 2인 데이터를 Write시에 저장 되는 노드들을 나타내고 있다.



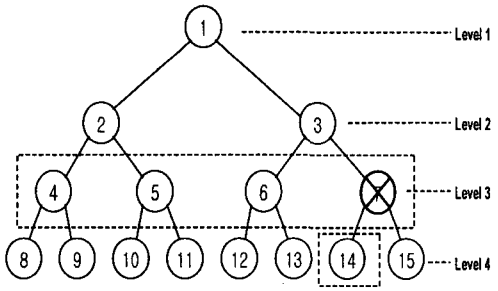
(그림 2) 레벨 2인 데이터의 저장되는 노드의 위치.

아래와 같은 트리를 Write알고리즘의 IDS로 일반화해 보면,

- 트리의 최대 레벨 : n
- 자식노드의 수 : t
- 레벨 L인 데이터의 분할개수 : t^{L-1}
- 노드에 저장되는 개수 : $t^{L-1} + t^L = (t+1)t^{L-1}$

따라서 $(t^{L-1}, (t+1)t^{L-1})$-IDS, L=1, 2, ⋯ , $n-1$ 인IDS로 표현할 수 있다.

위 IDS 표현에서 알 수 있듯이 레벨 단위 데이터 분할 프로토콜의 모든 IDS는 $t+1$인 확장율을 갖으며, 확장율이 $t+1$인 Class에 속한다. 그리고 Section 2의 정리2에 의해 데이터의 레벨이 증가 할수록 데이터의 가용성이 증가하는 것을 알 수 있다. 보안성 측면에서 단순히 분할로서 보안성 정도를 나타내기란 어렵지만 분할이 많을수록 보안성이 증가한다고 가정하면 데이터의 레벨이 증가 할수록 분할되는 개수가 증가하기 때문에 보안성이 증가하는 것을 알 수 있다. 또한 네트워크 설계시 확장율 e가 주어진다면 $e-1$개의 자식 노드를 갖는 트리 구조를 만들면 된다는 것을 알 수 있다.



(그림 3) 레벨 3인 데이터 Read 시의 경우.

지금까지 데이터를 Write 경우에 대해 알아 보았다. 이번에는 Read의 기능에 대해 알아보면, Read의 기능은 Write의 기능 보다 더 간단하다. Read할 경우, Read할 데이터의 레벨과 일치하는 트리의 레벨의 노드들만 읽으면 된다. 만일 Read시에 임의의 노드가 실패할 경우 그 노드의 자식 노드 중 하나를 Read하면 된다. 이와 같이 Read의 기능은 높은 에러 복구율을 나타낸다. (그림 3)은 레벨 3인 데이터를 Read하는 경우로 7번 노드가 실패시 복구하기 위해 그 자식 노드인 14번 노드를 Read하는 예이다.

Read Operation은 최신의 데이터를 Read하고 Write Operation은 최신의 데이터에 업데이트 해야 한다. 이러한 동작들이 제대로 이루어지기 위해서는 몇 가지 제약이 필요하다. 여러 개의 Read와 Write Operation들이 서로 다른 데이터들에 대한 동시 실행에는 상관은 없겠지만, 동일한 데이터에 대한 동시 실행에는 다음과 같은 제약이 필요하다.

- 여러 개의 Read Operation이 실행이 가능함.
- 하나의 Write Operation만 실행이 가능함.
- Read와 Write Operation들 중 하나만이 실행이 가능함.

4. 가용성(availability) 평가

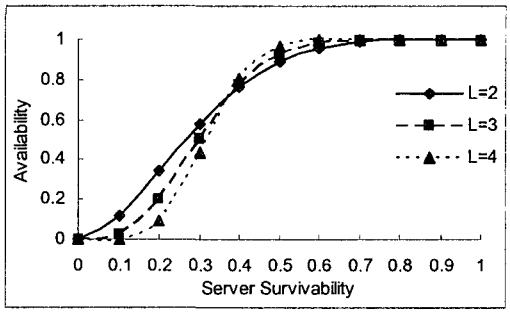
가용성을 원활하게 구하기 위해 다음과 같은 가정을 한다.

- 가정 1

  서버들의 실패는 독립적이며, 모두 다 같은 성공율을 갖는다.

(m, n)-IDS는 Read의 경우 $n-m$의 서버 실패를 묵인할 수 있다고 했다. 시스템의 서버들이 트리 구조가 아닌 각각의 노드가 독립된 구조인 경우, 가용성 측정은 Section 2의 식 (1)에 의해 구할 수 있다[5].

(그림 4)는 $t=2$인 트리 구조에서 적용되는 IDS들을 독립된 구조에서 Read의 가용성 측정 결과를 나타내고 있다. (그림 4)에서 나타나듯 레벨이 증가할수록 가용성이 증가하는 것을 알 수 있다.



(그림 4) 독립된 구조에서의 IDS들의 Read 가용성.

다음은 트리 구조에서의 Read의 가용성을 알아보기 위해 공식을 유도해 보자. 먼저 임의의 자식 노드를 갖는 노드 하나를 Read할 경우를 생각해보면 자식 노드가 2일 때의 가용성은 다음과 같다.

$$P_s+(1-P_s)\sum_{i=1}^{2}\binom{2}{i}P_s^i(1-P_s)^{2-i}$$

자식 노드가 3일 때는,

$$P_s+(1-P_s)\sum_{i=1}^{3}\binom{3}{i}P_s^i(1-P_s)^{3-i},$$

4 일 때는,

$$P_s+(1-P_s)\sum_{i=1}^{4}\binom{4}{i}P_s^i(1-P_s)^{4-i},$$

⋮

따라서 자식 노드가 t일 때는

$$P_s+(1-P_s)\sum_{i=1}^{t}\binom{t}{i}P_s^i(1-P_s)^{t-i}$$

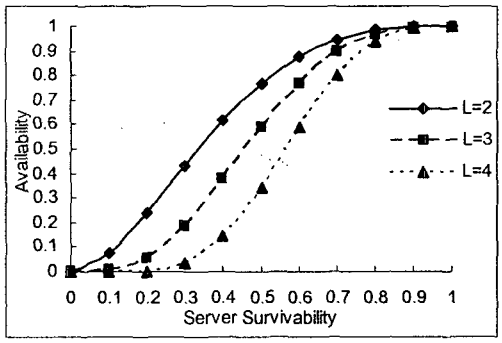
이라는 것을 알 수 있다. 실제 트리에서 레벨 L인 데이터를 Read할 때 t^{L-1}개를 먼저 Read하기 때문에 트리의 최대 레벨이 k인 레벨 단위 데이터 분할 프로토콜에서 Read의 가용성은 다음 공식에 의해 구할 수 있다.

$P(1, 1)=P_s,\ L=1$

$$P(m, n)=\left(P_s+(1-P_s)\sum_{i=1}^{t}\binom{t}{i}P_s^i(1-P_s)^{t-i}\right)^{L-1},\ L=2, \cdots k-1$$

(그림 5)는 $t=2$인 트리 구조에서의 레벨별 IDS의 Read 가용성을 나타내고 있다.
(그림 5)에서 볼 수 있듯이 Read 가용성이 레벨이 증가할수록 떨어지는 것을 알 수 있다. 그 이유는 트리 구조에서 부모, 자식 노드간의 관계가 독립적인 관계가 아니기 때문이다. 즉, 레벨이 L인 노드를 Read할

경우 레벨이 $L-1$인 노드들은 부모 노드인 레벨 L의 노드들의 성공여부에 영향을 받으며, 레벨이 증가 할수록 그러한 노드들이 증가하기 때문이다. 따라서 트리 구조에서의 Read의 가용성은 실제 데이터의 가용성이 아니다. 독립된 구조에서의 Read의 가용성이 시스템에서의 실질적인 데이터의 가용성이라 할 수 있다.

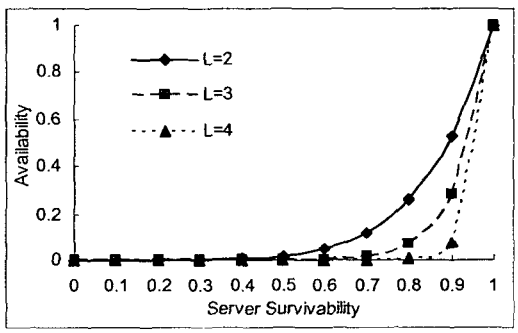


(그림 5) 트리 구조에서의 IDS들의 Read 가용성.

한편 Write의 가용성에 대해 알아보면, 레벨별 Write 가용성은 독립된 구조와 트리의 구조가 동일하다. 그 이유는 Write할 노드 수가 같고, 각각의 노드의 성공 여부만 달려있기 때문이다. Write의 가용성은 레벨이 증가할수록 Write할 노드 수가 증가하기 때문에 떨어진다. 가용성은 다음 식에 의해 구할 수 있다.

$P(m, n) = P_s,\ L=1$

$P(m, n) = (P_s \cdot P_s')^{L-1} = (P_s^{n+1})^{L-1},\ L=2, \cdots k-1$



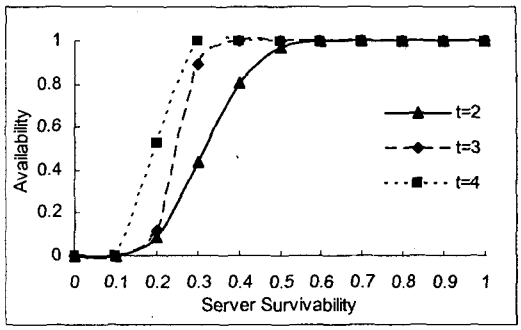
(그림 6) 독립구조 및 $t=2$ 트리 구조의 Write 가용성.

(그림 6)은 $t=2$인 트리 구조의 Write의 가용성을 나타내고 있다.

지금까지는 자식 노드 수가 일정한 상태에서 가용성과 레벨간의 관계에 대해서 알아보았다. 이번에는 가용성과 자식 노드 수와의 관계, 즉, 가용성과 확장율의 관계에 대해 알아보자. 레벨 단위 데이터 분할 프로토콜은 자식수가 증가 할수록 Read의 가용성은 증가하나 Write의 가용성은 감소한다. (그림 7)은 레벨이 2로 일정한 상태에서 자식의 노드 수의 변화에 대한 Read의 가용성을 나타내고 있다.

5. 결론

본 논문은 Survivable Storage System에서 시스템



(그림 7) 자식 노드수의 변화에 대한 Read의 가용성.

성능, 데이터의 가용성 및 안전성의 관점에서 데이터를 효율적으로 관리하기 위한 프로토콜을 제안하였다. 그리고 가용성 측정을 통해 레벨 단위 데이터 분할 프로토콜은 데이터 레벨이 증가할수록 트리 구조에서의 Read 및 Write 가용성은 떨어지지만, 실질적인 데이터의 가용성 및 보안성은 증가한다는 것을 보았다.

앞으로 계획으로서 이 레벨 단위 데이터 분할 프로토콜에서 최대 레벨의 IDS가 가장 최적인 IDS가 되는 구조에 대해서 연구할 예정이며, 트리 구조의 확장 및 축소 즉, 네트워크의 변형에 대해서도 연구할 예정이다.

참고문헌

[1] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G.. R. Ganger, H. Kiliccote, P. K. Khosla, "Survivable Information Storage Systems", IEEE Computer, 33(8), 2000 August, pp 61-68.

[2] J. J. Wylie, M. Bakkalogu, V. Pandurangan, M. W. Bigrigg, S Oguz, K. Tew, C. Williams, G. R. Ganger, P. K. Khosla, "Selecting the Right Data Distribution Scheme for a Survivable Storage Systems", Technical Report CMU-CS-01-120 Carnegie Mellon University, 2001 May.

[3] H.M. Sun, S.P. Shieh, "Optimal Information-Dispersal for Increasing the Reliability of a Distributed Service", IEEE Transactions on Reliability, vol 46, NO. 4, 1997 Dec.

[4] L. Gong, "Increasing availability and security of an authentication service", IEEE J. Selected Areas in Communication, vol 11, 1993 Jun, pp 657-662.

[5] D. P. Siewiorek, R. S. Swarz, "Reliable Computer Systems: design and evaluation", Digital Press, Second Edition, 1992.

[6] M.O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance", J. ACM, vol 36, 1989 Apr, pp 335-348.

[7] C. Asmuth, G. R. Blackley, "Pooling Splitting and Restituing Information to overcome total failure of some channels of communication", IEEE Proc. 1982 Symp. Security & Privacy, 1982, pp 156-169.